

Appl. No. 10/697,911  
Response dated 09/01/2006  
Reply to Office Action of 06/02/2006

**REMARKS**

In this Office Action, the Examiner stated that the IDS filed on 04/24/2003 failed to comply with 37 CFR 1.97. Claims 3, 5, 6, 7 and 14 were rejected under 35 U.S.C. §112, first paragraph for indefiniteness. Claims 1, 15, 26 and 29 were rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter. Claims 1, 2, 4, 8 – 13 and 15 - 29 were rejected under 35 U.S.C. §102(b) as being anticipated by Edberg.

Examiner Oni and Examiner Leroux are greatly thanked for the interview on August 31, 2006. In that interview, Claim 1, the applied reference as well as the 101 and 102 rejections were discussed. No agreements were reached.

Filed concurrent with this Response is an Information Disclosure Statement (IDS) containing an English Abstract of Japanese patent JP9218867. Applicants believe that the present IDS falls within 37 C.F.R. 1.97. Further, Applicants submit that a fee for the IDS need not be paid since the IDS had previously been submitted, albeit the reference therein was not in English.

The Examiner rejected Claims 3, 5 - 7 and 14 under 35 U.S.C. §112 as being indefinite because of the "IF" in those claims. Applicants amended the claims to ensure that they are no longer indefinite.

In response to the 35 U.S.C. §101 rejection of Claims 1, 15, 26 and 29, Applicants amended Claims 1 and 14 to include that "data having a fixed format are being edited on a computer system that uses data having a non-fixed format and that after being edited, the data continues to have the fixed format" and "editing the fixed-length statement." Applicants also amended Claim 26 to include the limitations "first computer system for transferring data to a second computer system, the first computer system using non-fixed-length data format and the second computer system using fixed-length data format, the first computer system" as well as "means for transferring the converted, edited substring to a second computer." Accordingly, Applicants submit that Claims 1, 14 and 26 are now directed toward statutory matter. Note that Claims 9, 15 - 25

CA920030001US1

Appl. No. 10/697,911  
Response dated 09/01/2006  
Reply to Office Action of 06/02/2006

and 29 are canceled. Thus, the 101 rejection of Claims 15 and 29 becomes moot. Claims 1 – 8, 10 – 14 and 26 – 28 were also amended to better claim the invention and new Claims 30 – 36 are presented for consideration.

By this amendment, Claims 1 – 8, 10 – 14, 26 – 28 and 30 – 36 are pending in the Application. For the reasons stated more fully below, Applicants submit that the pending claims are allowable over the applied reference. Hence, reconsideration, allowance and passage to issue are respectfully requested.

As disclosed in the SPECIFICATION, files encoded in EBCDIC having both fields and statements of fixed byte length may be downloaded to a workstation implementing Unicode for revision. When the file is downloaded, the file content is converted from EBCDIC to Unicode. Conversely, when the file is uploaded from the workstation to the legacy system, the file content is converted from Unicode to EBCDIC. Typically, prior art conversion methods that convert from EBCDIC to Unicode and from Unicode to EBCDIC are unaware of statements and are, therefore, unaware of the length of statements within the file. Although the same statement represented in Unicode on the workstation has the same number of characters, it may have a different byte length because the characters are represented differently since each Unicode character may have a different byte length than its EBCDIC equivalent (i.e., if encoded in UTF-8, UTF-16 or UTF-32).

Thus on the workstation, each character is displayed as one Unicode character but because a Unicode character may be the equivalent of multiple bytes, it may not be interpreted correctly in a mixed EBCDIC encoding of a legacy system. An editing program, moreover, may extract fields from a statement, modify the fields, and reassemble the individual fields to form a new statement. In today's world of graphical user interfaces, an editing program may display each field of a programming statement in a different color. In each of these cases, the editor needs to know, based on the number of bytes in the field, which group of Unicode characters form a field.

CA920030001US1

Page 17 of 21

Appl. No. 10/697,911  
Response dated 09/01/2006  
Reply to Office Action of 06/02/2006

Current Unicode string manipulation classes assume that lengths are defined as a number of Unicode characters. This assumption is wholly inadequate for the case cited above, i.e., when a statement in the file is altered through insertion, deletion, or replacement of characters on a Unicode workstation before the file is converted back to EBCDIC. Thus, the statement length may change from the original fixed statement length resulting in invalid statements. Therefore, the industry needs a new Unicode string manipulation class in which lengths are defined as a number of bytes in the legacy code page encoding, and the length of fields and statements remain constant.

Consequently, the present invention provides a method by which a fixed-length statement may be properly edited on a computer system that uses non-fixed-length statement or data. Accordingly the invention receives a data byte array that is to be used to edit the fixed-length statement. Upon receiving the data byte array, an encoding for the data byte array is determined by determining a number of bytes in each of a plurality of fixed-length fields that comprise the fixed-length statement. Then, a number of bytes in the fixed-length statement is determined before the a data string is created from the data byte array. In this case, the data string will be encoded using the determined encoding and an attribute is assigned to each byte of the data string.

The invention is set forth in claims of varying scopes of which Claim 1 is illustrative.

1. A computer-implemented method of editing data having a fixed format on a computer system that uses data having a non-fixed format such that after being edited, the data continues to have the fixed format, the method comprising the steps of:

*receiving, by the computer system, a data byte array, the data byte array having the non-fixed format and including data for editing a fixed-length statement;*

*determining the fixed format encoding of the of the fixed-length statement by determining a number*

CA920030001US1

Page 18 of 21

Appl. No. 10/697,911  
Response dated 09/01/2006  
Reply to Office Action of 06/02/2006

*of bytes in each of a plurality of fixed-length fields that comprise the fixed-length statement;  
determining a number of bytes in the fixed-length statement;  
creating a data string from the data byte array, the data string being encoded using the determined fixed format encoding, given a starting byte position and the number of bytes in the fixed-length statement;  
assigning an attribute to each byte of the data string; and  
editing the fixed-length statement using the data string.* (Emphasis added.)

The Examiner rejected the claims under 35 U.S.C. §102(b) as being anticipated by Edberg et al. Applicants respectfully disagree.

Edberg et al. purport to teach a Unicode converter. Specifically, Edberg et al. provide a Unicode conversion technique that ensures that resulting character codes are interchangeable with other platforms. The Unicode conversion system maps a single Unicode character or a sequence of characters to either a single target character or a sequence of target characters. With round trip fidelity, Unicode text can be converted to target text and then back again to Unicode original text. The interchangeability is ensured by maximizing the use of standard target characters and by minimizing the use of private characters.

The Examiner stated that in col. 7, lines 44 – 63, Edberg et al. teach the step of *receiving a first data byte array*. Applicants respectfully disagree.

The passage in col. 7, lines 44 – 63 is directed toward the definition of the word "DEFAULT" as used by Edberg et al. in their disclosure. However, nowhere in the passage is there a mention of the phrase "selecting a first data byte array" or an inference of it. Applicants submit that Edberg et al. do not show the step of *receiving a first data byte array*.

The Examiner further stated that in col. 14, line 52 to col. 15, line 15, Edberg et al. disclose the step of *determining an encoding for the data byte*

CA920030001US1

Page 19 of 21

Appl. No. 10/697,911  
Response dated 09/01/2006  
Reply to Office Action of 06/02/2006

***array by determining a number of bytes in each of a plurality of fixed-length fields that comprise the fixed-length statement.*** Again, Applicants disagree.

The passage states that if data is to be converted, an offset array is updated. The offset array is an array of offsets (pointers) associated with an input string that indicate where certain changes such as font changes, line breaks, language changes, etc., occur within the input string that the calling application deems significant. Updating the offset array involves adjusting the offsets (pointers) for different length characters such that the offsets point to the corresponding character in the target encoding.

Thus, the passage does not mention that encoding for the data byte array occurs ***by determining a number of bytes in each of a plurality of fixed-length fields that comprise the fixed-length statement*** as claimed. Indeed, Edberg et al. does not even mention the term **fixed-length fields** or the term **fixed-length statement**.

Hence, Applicants maintain that Edberg et al. do not teach the step of ***determining an encoding for the data byte array by determining a number of bytes in each of a plurality of fixed-length fields that comprise the fixed-length statement.***

In addition, the Examiner stated in col. 14, line 52 to col. 15, line 15, Edberg et al. disclose the step of ***determining a number of bytes in the fixed-length statement.*** But as mentioned immediately above, Edberg et al. do not teach the term **fixed-length statement** and therefore cannot teach the step of ***determining a number of bytes in the fixed-length statement.***

The claim elements that start with "creating ..." and "editing ..." contain each the term ***fixed-length statement.*** Consequently, they too are not taught by Edberg et al.

Finally, the Examiner stated that in col. 22, lines 30 – 57, Edberg et al. disclose the step of ***assigning an attribute to each byte of the first data string.*** Once again, Applicants disagree.

CA920030001US1

Page 20 of 21

Appl. No. 10/697,911  
Response dated 09/01/2006  
Reply to Office Action of 06/02/2006

In col. 22, lines 30 – 57, Edberg et al. stated that a desired attributes mask is formatted like an actual attributes mask, but sets bits depending on which of the attributes is important to obtain a correct mapping for a particular table and variant. But note that it is a mask that is used to obtain a correct mapping. Edberg et al. do not teach **assigning an attribute to each byte of the first data string** as claimed.

Consequently, Applicants submit that the Claim 1, as well as its dependent claims, is allowable over the cited reference. The other independent claims (i.e., Claims 14, 26 and 30) and their dependent claims, which all incorporate the **emboldened-italicized** limitations of the above-reproduced Claim 1 is also allowable. Consequently, Applicants once more respectfully request reconsideration, allowance and passage to issue of the claims in the application.

Respectfully Submitted

By:

Volel Emile  
Attorney for Applicants  
Registration No. 39,969  
(512) 306-7969

CA920030001US1

Page 21 of 21